

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 02-027389

(43)Date of publication of application : 30.01.1990

(51)Int.Cl.

G09C 1/00

(21)Application number : 63-176775

(71)Applicant : SONY CORP

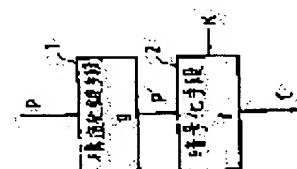
(22)Date of filing : 15.07.1988

(72)Inventor : NODA SHIGETOSHI

**(54) ENCIPHERING METHOD AND ENCIPHERING DEVICE/DECODING DEVICE USING ENCIPHERING METHOD CONCERNED****(57)Abstract:**

**PURPOSE:** To eliminate the danger of decoding by strengthening the cipher strength by only adding a circuit for executing a processing of a simple algorithm to strong enciphering/decoding circuits of a DES system, an FEAL system, etc.

**CONSTITUTION:** A structured key 1 makes a code  $p'$  from a plain sentence ( $p$ ) from an algorithm ( $g$ ), and ( $g$ ) is a secret. An enciphering means 2 makes a cipher text ( $c$ ) by executing strongly an encipherment by an algorithm ( $f$ ) by using the code  $p'$  and a key ( $k$ ). ( $f$ ) can be disclosed. A decoding means 3 makes a code from the cipher text ( $c$ ) by an algorithm  $f^{-1}$  by using ( $k$ ), and a key means 4 decodes the plain sentence ( $p$ ) from  $p'$ . In this case, conditions of  $f = f^{-1}$ , random correspondence  $f(p', k) = \text{Rand}(p', k)$ ,  $f(., p', ., k) = \text{Rand}(., p', ., k)$ , encipherment  $c = f(p')$ , and decoding  $p' = f^{-1}(c)$  are satisfied. When  $p'$  and ( $c$ ) are known, ( $k$ ) can be decoded by a detailed inspection, but when one of them is unknown, decoding is impossible. ( $f$ ) is obtained by using DES and FEAL systems. Also,  $g = g^{-1}$ , and a bit transposition of a simple structure is used. According to this constitution, decoding becomes impossible even in the case of a high speed operation processing.

**LEGAL STATUS**

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision]

of rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

## ⑫ 公開特許公報(A) 平2-27389

⑤ Int. Cl.<sup>5</sup>

識別記号

庁内整理番号

⑬ 公開 平成2年(1990)1月30日

G 09 C 1/00

7368-5B

審査請求 未請求 請求項の数 6 (全10頁)

⑭ 発明の名称 暗号化方法及び該暗号方法を用いた暗号化装置／復号化装置

⑯ 特 願 昭63-176775

⑰ 出 願 昭63(1988)7月15日

⑱ 発 明 者 納 田 重 利 東京都品川区北品川6丁目7番35号 ソニー株式会社内

⑲ 出 願 人 ソニー株式会社 東京都品川区北品川6丁目7番35号

⑳ 代 理 人 弁理士 杉浦 正知

## 明 細 書

## 1. 発明の名称

暗号化方法及び該暗号方法を用いた暗号化装置／復号化装置

## 2. 特許請求の範囲

(1) 入力コードを $x$ 、キーコードを $k$ 、任意の入力コードのビット変化高を $\Delta x$ 、任意のキーコードのビット変化高を $\Delta k$ としたとき、

$$f = f^{-1} \text{ 又は } f^{-1} \text{ が存在し、}$$

$$f(x, k) = \text{Rand}(x, k)$$

$$f(\Delta x, \Delta k) = \text{Rand}(\Delta x, \Delta k)$$

を満足するようなアルゴリズム $f$ の暗号化を行う際の鍵として、上記アルゴリズムの鍵と独立で且つ鍵自体が暗号化アルゴリズムをなす構造化鍵を用いるようにした暗号化方法。

(2) 入力コードを $x$ 、キーコードを $k$ 、任意の入力コードのビット変化高を $\Delta x$ 、任意のキーコードのビット変化高を $\Delta k$ としたとき、

$$f = f^{-1} \text{ 又は } f^{-1} \text{ が存在し、}$$

$$f(x, k) = \text{Rand}(x, k)$$

$$f(\Delta x, \Delta k) = \text{Rand}(\Delta x, \Delta k)$$

を満足するようなアルゴリズム $f$ の暗号化を行う暗号化装置と直列に、独立な構造化鍵のアルゴリズムの処理回路を配置するようにした暗号化装置。

(3) 入力コードを $x$ 、キーコードを $k$ 、任意の入力コードのビット変化高を $\Delta x$ 、任意のキーコードのビット変化高を $\Delta k$ としたとき、

$$f = f^{-1} \text{ 又は } f^{-1} \text{ が存在し、}$$

$$f(x, k) = \text{Rand}(x, k)$$

$$f(\Delta x, \Delta k) = \text{Rand}(\Delta x, \Delta k)$$

を満足するようなアルゴリズム $f$ の暗号化を行う暗号化装置と直列に、独立な構造化鍵のアルゴリズムの処理回路を配置するようにした復号化装置。

(4) 入力コードを $x$ 、キーコードを $k$ 、任意の入力コードのビット変化高を $\Delta x$ 、任意のキーコードのビット変化高を $\Delta k$ としたとき、

$$f = f^{-1}$$

$$f(x, k) = \text{Rand}(x, k)$$

$$f(\Delta x, \Delta k) = \text{Rand}(\Delta x, \Delta k)$$

を満足するような暗号化アルゴリズム $f$ の処理を

**THIS PAGE BLANK (USPTO)**

行う前後に、少なくとも互いに逆関数となる構造を持つ独立な暗号化アルゴリズムの処理を設けるようにした暗号化方法。

(5) 入力コードを $x$ 、キーコードを $k$ 、任意の入力コードのビット変化高を $\Delta x$ 、任意のキーコードのビット変化高を $\Delta k$ としたとき、

$$\begin{aligned} f &= f^{-1} \\ f(x, k) &= \text{Rand}(x, k) \\ f(\Delta x, \Delta k) &= \text{Rand}(\Delta x, \Delta k) \end{aligned}$$

を満足するような暗号化アルゴリズム $f$ の暗号化回路の前段と後段に、少なくとも互いに逆関数となる構造を持つ独立な暗号化アルゴリズムの暗号化回路を設けるようにした暗号化装置。

(6) 入力コードを $x$ 、キーコードを $k$ 、任意の入力コードのビット変化高を $\Delta x$ 、任意のキーコードのビット変化高を $\Delta k$ としたとき、

$$\begin{aligned} f &= f^{-1} \\ f(x, k) &= \text{Rand}(x, k) \\ f(\Delta x, \Delta k) &= \text{Rand}(\Delta x, \Delta k) \end{aligned}$$

を満足するような暗号化アルゴリズムの復号化回

路を設けるようにした暗号化装置。られる暗号化方法及びその暗号化装置／復号化装置において、DES方式やFEAL方式のように強い暗号化を行うアルゴリズムの前段と後段に、弱い暗号化アルゴリズムを付加することにより、暗号強度を強化するようにしたものである。

#### 〔従来の技術〕

暗号化技術は、アタッカーにより情報が盗用或いは漏洩されることを防止するとともに、相互認証の基で確実に通信し合うことを可能にする。本格的に到来するであろう大規模分散情報通信ネットワークシステムにおいて、情報を保護していくために、このような暗号化技術の発達とその普及は不可欠である。

暗号方式には、大別して慣用鍵方式（共通鍵方式）と公開鍵方式とがある。慣用鍵方式では、暗号化鍵と復号化鍵とが共通とされる。公開鍵方式では、暗号化鍵と復号化鍵とが異なり、暗号化鍵が公開される。

暗号方式には、種々の方式が提案されている。

路の前段と後段に、少なくとも互いに逆関数となる構造を持つ暗号化アルゴリズムの復号化回路を設けるようにした復号化装置。

#### 3. 発明の詳細な説明

##### 〔産業上の利用分野〕

この発明は、コンピュータネットワークシステムで通信されるデータの保護のために用いられる暗号化方法及びその暗号化装置／復号化装置に関する。

##### 〔発明の概要〕

この発明は、コンピュータネットワークシステムで通信されるデータの保護のために用いられる暗号化方法及びその暗号化装置／復号化装置において、DES方式やFEAL方式のように強い暗号化を行うアルゴリズムと直列に、鍵自体が暗号化アルゴリズムの構造化鍵を付加することにより、暗号化強度を強化するようにしたものである。

また、この発明は、コンピュータネットワークシステムで通信されるデータの保護のために用い

その中で実用的な符号としては、慣用鍵方式の暗号化方式においては、DES (Data Encryption Standard) 方式と、FEAL (Fast Data Encipherment Algorithm) 方式があり、公開鍵暗号化方式においては、RSA (Rivest Shamir Adleman) 方式がある。

DES方式やFEAL方式は、強い暗号化を行うアルゴリズムにより暗号化が行われるため、乱み潰し以外に解読できないとされている。このような強い暗号化方式では、入力される平文と出力される暗号文及び鍵と出力される暗号文との関係がランダムに結ぶものと言える。このような関係は、入力される平文を $p$ 、鍵を $K$ とした時、

$$f(p, K) = \text{Rand}(p, K)$$

と表せる。

更に、このような強い暗号化方式では、入力される平文のビット変化高に対する出力される暗号文のビット変化高及び鍵のビット変化高に対する出力される暗号文のビット変化高がランダムであり、

**THIS PAGE BLANK (USPTO)**

$f(\Delta p, \Delta K) = \text{Rand}(\Delta p, \Delta K)$   
 であるように構成されている。

一般に、このような暗号化方式のアルゴリズムは、インボリューション構造を持つ。インボリューション構造は、

$$c = f(p), \quad p = f^{-1}(c),$$

において

$$f = f^{-1}$$

が成立する構造である。インボリューション構造を持つ場合には、暗号化と復号化が同様の処理プロセスで行える。EX-ORをとるアルゴリズム(mod 2の加算を行うアルゴリズム)は、インボリューション構造の簡単な例である。

(発明が解決しようとする課題)

暗号化方式としては、上述したように、種々のものが提案されている。そして、信頼性の保証とハードウェアの共通化をはかれるために、暗号化方式を標準化することが検討されている。

ところが、超並列処理コンピュータの開発等、

FEAL方式では51万年程かかることになり、DES方式やFEAL方式は、十分安全な暗号といえる。

ところが、近年のコンピュータ処理速度の向上は目覚ましく、 $10^8$ の並列処理で演算を行うことが実現可能になった場合には、乱み潰して解読を行うのに、DES方式では9.6時間程、6.1カ月程で良いことになる。

したがって、この発明は、このような超並列型のコンピュータの開発等、将来の演算速度の向上に備えて、より安全な暗号化を行える暗号化装置の提供を目的とする。

(課題を解決するための手段)

この発明は、入力コードをx、キーコードをk、任意の入力コードのビット変化高を $\Delta x$ 、任意のキーコードのビット変化高を $\Delta k$ としたとき、

$$f = f^{-1} \text{ 又は } f^{-1} \text{ が存在し、}$$

$$f(x, k) = \text{Rand}(x, k)$$

$$f(\Delta x, \Delta k) = \text{Rand}(\Delta x, \Delta k)$$

コンピュータ技術の発展により、上述した従来の暗号化方式では、十分安全であるとは言えなくなっている。このことが、暗号化方式を標準化していく上での一つの障害となっている。

つまり、暗号の強さのひとつのバロメータとして鍵のビット長がある。すなわち、鍵のビット長が長くなればなるほど、乱み潰して解読される危険性が少なくなる。したがって、鍵のビット長を増加していくことで、暗号強度を増加できる。

ところが、鍵のビット数が増え、処理が複雑化して、コストパフォーマンスが悪くなる。適当な妥協点として、DES方式やFEAL方式では、従来、ビット長を64ビットとしている。

このようにした場合、例えば、DES方式では、乱み潰して暗号を解読するのに、 $2^{56}$ 回の演算が必要である。FEALでは、乱み潰して暗号を解読するのに、 $2^{64}$ 回の演算が必要である。したがって、並列処理を行わず、 $1\mu s$ で1回の速度で演算を行って解読する場合には、乱み潰して解読を行うのに、DES方式では11000年程かかり、

を満足するようなアルゴリズムfの暗号化を行う際の鍵として、上記アルゴリズムの鍵と独立で且つ鍵自体が暗号化アルゴリズムをなす構造化鍵を用いるようにした暗号化方法である。

また、入力コードをx、キーコードをk、任意の入力コードのビット変化高を $\Delta x$ 、任意のキーコードのビット変化高を $\Delta k$ としたとき、

$$f = f^{-1} \text{ 又は } f^{-1} \text{ が存在し、}$$

$$f(x, k) = \text{Rand}(x, k)$$

$$f(\Delta x, \Delta k) = \text{Rand}(\Delta x, \Delta k)$$

を満足するようなアルゴリズムfの暗号化を行う暗号化装置と直列に、独立な構造化鍵のアルゴリズムの処理回路を配置するようにした暗号化装置である。

また、入力コードをx、キーコードをk、任意の入力コードのビット変化高を $\Delta x$ 、任意のキーコードのビット変化高を $\Delta k$ としたとき、

$$f = f^{-1} \text{ 又は } f^{-1} \text{ が存在し、}$$

$$f(x, k) = \text{Rand}(x, k)$$

$$f(\Delta x, \Delta k) = \text{Rand}(\Delta x, \Delta k)$$

**THIS PAGE BLANK (USPTO)**



を満足するようなアルゴリズム  $f$  の暗号化を行う暗号化装置と直列に、独立な構造化鍵のアルゴリズムの処理回路を配置するようにした復号化装置である。

また、入力コードを  $x$ 、キーコードを  $k$ 、任意の入力コードのビット変化高を  $\Delta x$ 、任意のキーコードのビット変化高を  $\Delta k$  としたとき、

$$f = f^{-1}$$

$$f(x, k) = \text{Rand}(x, k)$$

$$f(\Delta x, \Delta k) = \text{Rand}(\Delta x, \Delta k)$$

を満足するような暗号化アルゴリズム  $f$  の処理を行う前後に、少なくとも互いに逆関数となる構造を持つ独立な暗号化アルゴリズムの処理を設けるようにした暗号化方法である。

また、入力コードを  $x$ 、キーコードを  $k$ 、任意の入力コードのビット変化高を  $\Delta x$ 、任意のキーコードのビット変化高を  $\Delta k$  としたとき、

$$f = f^{-1}$$

$$f(x, k) = \text{Rand}(x, k)$$

$$f(\Delta x, \Delta k) = \text{Rand}(\Delta x, \Delta k)$$

また、DES方式やFEAL方式のように強い暗号化を行うアルゴリズムの前段と後段に、弱い暗号化アルゴリズムを付加することにより、暗号強度が強化される。

#### 〔実施例〕

この発明の実施例について、以下の順序にしたがって説明する。

#### a. 構造化鍵を用いて暗号強度を強化させる例

##### a 1. 構造化鍵について

a 2. 構造化鍵としてEX-ORを用いた場合の考察

a 3. 構造化鍵としてビット転置器を用いた場合の考察

##### a 4. 構造化鍵を用いた場合の実施例

#### b. 強い暗号化処理手段の前段と後段に弱い暗号化手段を配して、暗号化強度を強化させる例

b 1. 強い暗号化処理手段の前段と後段に弱い暗号化手段を配して、暗号化強度を強化させる例の基本構成

を満足するような暗号化アルゴリズム  $f$  の暗号化回路の前段と後段に、少なくとも互いに逆関数となる構造を持つ独立な暗号化アルゴリズムの暗号化回路を設けるようにした暗号化装置である。

また、入力コードを  $x$ 、キーコードを  $k$ 、任意の入力コードのビット変化高を  $\Delta x$ 、任意のキーコードのビット変化高を  $\Delta k$  としたとき、

$$f = f^{-1}$$

$$f(x, k) = \text{Rand}(x, k)$$

$$f(\Delta x, \Delta k) = \text{Rand}(\Delta x, \Delta k)$$

を満足するような暗号化アルゴリズムの復号化回路の前段と後段に、少なくとも互いに逆関数となる構造を持つ暗号化アルゴリズムの復号化回路を設けるようにした復号化装置である。

#### 〔作用〕

DES方式やFEAL方式のように強い暗号化を行うアルゴリズムと直列に、鍵自体が暗号化アルゴリズムの構造化鍵を付加することにより、暗号化強度が強化される。

#### b 2. 強い暗号化処理手段の前段と後段に弱い暗号化手段を配して、暗号化強度を強化させる構成の実施例

##### a. 構造化鍵を用いて暗号強度を強化させる例

##### a 1. 構造化鍵について

現代暗号規約において、暗号に対する信用性や安全性をはかるために、以下のような規約が提案されている。

① 鍵以外の暗号化処理が公開されている。

② 解読者は、鍵以外の全ての情報を知っているとして、鍵の読み直し解読法又は計算量的安全な方法以外に解読法が知られていない。

このような規約を遵守した上で、より強い暗号化を行う方法を考察することにする。

従来の暗号化アルゴリズムでは、鍵が単純なコードだけである。より強い暗号化を行うために、これに加えて構造化された鍵を用いることが考えられる。

**THIS PAGE BLANK (USPTO)**

すなわち、DES方式やFEAL方式のように強い暗号化アルゴリズムと直列に、鍵自体が暗号化を行う構造の鍵を設けることが考えられる。このように、構造化された鍵を用いれば、解読がより困難になると考えられる。

このような構造化鍵は秘密にされる。そして、構造化鍵以外の全ての暗号化処理は、現代暗号化規約に則って公開される。構造化鍵は、記録媒体に記憶させ、秘密とされる関係上、単純な構造であることが望ましい。

第1図は、このような構造化鍵を用いた暗号化の処理を示すものである。第1図において、1は構造化鍵手段、2は暗号化手段である。構造化鍵手段1と暗号化手段2とが直列に配置される。

構造化鍵手段1は、アルゴリズムgにより、入力される明文pを暗号化し、中間コードp'を生成するものである。この構造化鍵1でどのようなアルゴリズムで暗号化がなされているかは、秘密とされる。

暗号化手段2には、構造化鍵手段1から出力さ

れる中間コードp'が与えられるとともに、鍵Kが与えられる。暗号化手段2は、中間コードp'を、鍵Kを用いて、アルゴリズムfにより強い暗号化を行い、暗号文cを生成するものである。この暗号化手段2のアルゴリズムは、公開することができる。

第2図は、このような構造化鍵を用いて暗号化された暗号文を復号する復号化の処理を示すものである。復号化は、第1図に示す暗号化の処理に対応している。

第2図において、3は復号化手段、4は構造化鍵手段である。復号化手段3は、鍵Kを用いて、アルゴリズムf'により暗号文cから中間コードp'を生成するものである。構造化鍵手段4には、復号化手段3で復号された中間コードp'が与えられる。構造化鍵手段4は、中間コードp'から明文pを生成するものである。これにより、暗号文cから明文pが解読される。

ここで、アルゴリズムf及びアルゴリズムgについて説明する。アルゴリズムfは、強い暗号化

を行うものである。このアルゴリズムfは、以下のような条件を満たしている。

①  $f = f^{-1}$  (インボリューション構造) 又は  $f^{-1}$  が存在する。

② 入力  $p'$ 、鍵  $K$  とすると、

$$f(p', K) = \text{Rand}(p', K)$$

つまり、 $p'$  や  $K$  に対して出力  $f(p', K)$  がランダムに対応する。

③ 入力のビット変化量を  $\Delta p'$ 、鍵のビット変化量を  $\Delta K$  とすると、

$$f(\Delta p', \Delta K) = \text{Rand}(\Delta p', \Delta K)$$

つまり、 $\Delta p'$  や  $\Delta K$  に対して、出力変化量もランダムに対応する。

④、暗号化は、

$$c = f(p')$$

で表せ、復号化は、

$$p' = f^{-1}(c)$$

で表せる。但し、インボリューション構造ならば、

$$p' = f(c) \text{ である。}$$

⑤ 入力  $p'$  や出力  $c$  が知られている時、鍵  $K$  の

乱み直し検査により鍵  $K$  は解読されるが、 $p'$  又は  $c$  の一方が未知の時、その一方と鍵  $K$  は解読不可能な構造をとる。

このようなアルゴリズム  $f$  としては、DES方式やFEAL方式を用いることができる。

構造化鍵手段1で行うアルゴリズム  $g$  は、以下のような構造を持つものである。

①  $g = g^{-1}$  又は  $g^{-1}$  が存在する。

②、入力  $p$  のビット変化量  $\Delta p$  に対して、

$$g(\Delta p) \neq \text{Rand}(\Delta p)$$

であっても良い。

③、入力  $p$  と出力  $p'$  の両方が知られた時、 $g$  の構造が知られても良いが、 $g$  の構造や  $g$  内の変数が知られない時、 $p$  と  $p'$  の一方が知られても、他方を知ることができない。

アルゴリズム  $g$  としては、極めて単純な構造、例えば、EX-OR構造、ビット転置、換字表等が考えられる。後述するように、構造化鍵として用いるアルゴリズム  $g$  としては、ビット転置が好ましく、EX-OR構造では、暗号化を十分強化

**THIS PAGE BLANK (USPTO)**

することはできない。

a 2. 構造化鍵として EX-OR を用いた場合の考察

強い暗号化を行う暗号化処理に対する構造化鍵として、全くランダムに選ばれた鍵  $k$  により EX-OR をとるアルゴリズムを採用した場合について考察する。

第3図は、構造化鍵のアルゴリズム  $g$  として、EX-OR をとる構造化鍵とした場合の例を示すものである。第3図において、11は構造化鍵手段であり、この場合には、この構造化鍵手段11は、鍵  $k$  と入力  $p$  との EX-OR をとる回路である。暗号化処理手段12は、DES方式やFEAL方式のような強い暗号化を行う暗号化回路である。

このように、EX-OR をとるアルゴリズムを、強い暗号化方法のアルゴリズムに対する構造化鍵とした場合の暗号強度について考察していくことにする。なお、便宜上  $f = f^{-1}$  とみなす。

このような暗号化は、平文を  $p$ 、EX-OR 回

路から出力される中間コードを  $p'$ 、暗号文を  $c$  とすると、

$$c = f(p', K), \quad p' = p \oplus k$$

で示される。また、復号化は、

$$p = k \oplus p', \quad p' = f(c, k)$$

であり、ここで、 $f$  はランダム関数である。上式をまとめると、暗号化は、

$$c = f(p \oplus k, K) = \text{Rand}(p \oplus k, K)$$

と表せ、復号化は、

$$p = k \oplus f(c, K) = k \oplus \text{Rand}(c, K)$$

と表せる。

上式から、以下のことがわかる。

すなわち、暗号化手続きは、2つの鍵  $k$  と  $K$  とのランダム関数によるものの、復号化手続きは、ひとつの鍵  $K$  のランダム関数によるもののため、復号化の方向では、解読において、 $K$  の漏み潰しは必要だが、 $k$  の漏み潰しを行う必要はない。

つまり、復号化の方向においては、

$$f(c, K) \oplus p = k = \text{一定}$$

という関係が成立する。このため、アタッカーは、

既知なる  $i$  個のサンプル ( $c_i, p_i$ ) を用いて、第4図に示すように、アルゴリズム  $f$  の処理を行う回路13と EX-OR をとる回路14とからなる構造化の処理回路によって鍵  $K$  を検索できる。すなわち、 $K$  の漏み潰しで、全てのサンプル  $i$  個において出力が一定となった時、その時の出力 ( $k$ ) と  $K$  が鍵として知られる。したがって、この手法の解読手数は、 $1 \cdot 2^B$  である ( $B$ : 鍵及び入出力コードのビット数)。

関数  $f$  は、ランダム関数であるため、出力 ( $k$ ) が1回一定となった時、その時の出力 ( $k$ ) と鍵 ( $K$ ) が本物でない確率 (誤り率) は、コード数  $2^B$  個内のひとつのコードをとる確率  $2^{-B}$  に同等な程低い。つまり、 $1 = 2$  でも十分解読可能と言える。

以上のような考察結果から、構造化鍵として、EX-OR をとるような構成では、十分に暗号強度を許可することにはならない。

a 3. 構造化鍵としてビット転置器を用いた場合

の考察

構造化鍵としてビット転置器を採用した場合について考察する。また便宜上  $f = f^{-1}$  とするものとする。

ビット転置器は、第5図に示すように、入力コードの各ビットをランダムに入れ替えるものである。ビット転置器のアルゴリズム  $g(p)$  は、

$$g(p) = \text{Rand}(p)$$

$$g(\Delta p) \neq \text{Rand}(\Delta p)$$

$$g \neq g^{-1}$$

として表せる。

第6図は、構造化鍵としてビット転置器を用いた場合の暗号化処理を示し、第7図は、その復号化処理を示している。

第6図に示す暗号化処理において、21は構造化鍵手段としてのビット転置器である。ビット転置器21により、アルゴリズム  $g$  により、入力のビットが転置される。22は暗号化手段である。暗号化手段22は、アルゴリズム  $f$  により、強い暗号化を行う。このアルゴリズムとしては、DE

**THIS PAGE BLANK (USPTO)**

S方式やFEAL方式が用いられる。

また、第7図に示す復号化処理において、23は復号化手段であり、復号化手段23でアルゴリズムfにより、暗号文cから中間コードp'が生成される。この中間コードp'が逆ビット転置器24に供給され、逆ビット転置器24でアルゴリズムg'により、逆ビット変換される。

この場合の暗号化は、暗号化手段22のアルゴリズムをf、ビット転置器21のアルゴリズムをg、入力される平文をp、中間コードをp'、暗号文をcとすると、

$$c = f(p', k) = f(g(p), K)$$

として表せる。また、復号化は、

$$p = g^{-1}(p') = g^{-1}(f(c, K))$$

として表せる。

fで示されるアルゴリズムは、強い暗号化を行っているので、入力変数のビット変化高( $\Delta p$ や $\Delta K$ 及び $\Delta c$ )に対して、出力がランダムに対応づけられるため、ビット変化高に対する解読攻撃に対しては、アルゴリズムfの鍵Kを虱み潰しに

より求め、アルゴリズムgの結線を虱み潰しにより求める以外に解読できない。また、アルゴリズムgは、その入力と出力に特別な関係が存在しないので、結局、ビット変化高の攻撃以外の攻撃に対しても、アルゴリズムfとアルゴリズムgの両方の検査以外に解読法はないと言える。

入力及び出力のビット数をBとすると、アルゴリズムfの鍵Kを虱み潰しで探す場合、 $2^B$ 回の演算が必要であり、アルゴリズムgの結線の虱み潰し回数はB!であるから、このようなアルゴリズムで暗号化を行った場合、虱み潰しで解読するのに、両方で $2^B \cdot B!$ 回の演算が必要になる。

$B = 64$ ビットとすると、

$$B! \approx 10^{99}$$

$$2^B \cdot B! \approx 10^{100}$$

となる。この場合には、1回の処理速度を $1 \mu s$ として、 $10^6$ の並列処理を行ったとしても、虱み潰しで解読を行うのに、約 $10^{94}$ 年必要になる。

#### a 4. 構造化鍵を用いた場合の実施例

第8図は、このように構造化鍵を用いて暗号強度を強化して通信を行う実施例を示すものである。

第8図において、通信を行うコンピュータシステム31には、構造化鍵手段32及びその構造化鍵手段32を管理するための鍵管理と、暗号/復号器の鍵管理を行う鍵管理手段33が設けられている。この構造化鍵手段32及び鍵管理手段33は、例えばソフトウェアで処理される。暗号/復号器35及びモデム36は、コントロールライン37を介してコンピュータシステム31と結ばれている。

コンピュータシステム31から回線34を介してデータを出力する場合には、コンピュータシステム31からのデータがこの構造化鍵手段32のアルゴリズムgにより、中間コードに変換される。この中間コードが暗号/復号器35に供給される。そして、暗号/復号器35により、アルゴリズムfにより暗号化され、この暗号化されたデータがモデム36を介して回線34に出力される。

回線34を介して伝えられてきたデータをコン

ピュータシステム31で受信する場合には、回線34を介して伝えられてきたデータがモデム36を介して暗号/復号器35に供給される。暗号/復号器35で、アルゴリズムfにより送られてきたデータが復号され、中間コードが生成される。

この中間コードがコンピュータシステム31の構造化鍵手段32に供給される。構造化鍵手段32で中間コードからデータが復号される。

b. 強い暗号化処理手段の前段と後段に弱い暗号化手段を配して、暗号化強度を強化させる例

b 1. 強い暗号化処理手段の前段と後段に弱い暗号化手段を配して、暗号化強度を強化させる例の基本構成

第9図は、DES方式やFEAL方式のような強い暗号化を行う暗号化手段の前段と後段に弱い暗号化を行う暗号化手段を配することにより、暗号強度を強化するようにした例である。なお、以下の説明では、 $f = f^{-1}$ とするものとする。第9図において暗号化手段51には、平文pが与えら

**THIS PAGE BLANK (USPTO)**



れる。暗号化手段31は、アルゴリズムhにより、弱い暗号化を行う。この暗号化手段51には、鍵kが与えられる。暗号化手段51からは、中間コード $p'$ が出力される。

暗号化手段51で生成された中間コード $p'$ が強い暗号化手段52に与えられる。強い暗号化手段52は、アルゴリズムfにより、強い暗号化を行う。強い暗号化手段52には、鍵kとは独立な鍵Kが与えられる。暗号化手段52から、暗号cが出力される。

強い暗号化手段52で暗号化されたコードは、更に、暗号化手段53に送られる。暗号化手段53は、アルゴリズム $h^{-1}$ により、弱い暗号化を行う。暗号化手段53からは、暗号 $c'$ が出力される。なお、アルゴリズムhはインボリューション構造で、 $h = h^{-1}$ である。

第10図は、その復号化処理を示している。第10図において、暗号文 $c'$ が復号化手段61に与えられる。復号化手段61で、アルゴリズムhにより、暗号文 $c'$ が復号され、暗号文cが復号

化手段61から出力される。

この暗号文cが強い復号化手段62に与えられる。強い復号化手段62は、アルゴリズムfにより、復号を行うものである。復号化手段62からは、中間コード $p'$ が出力される。この中間コード $p'$ が復号化手段63に与えられる。

ここで、アルゴリズムfは、以下のような条件を満たしている。

①  $f = f^{-1}$  (インボリューション構造)

② 入力 $p'$ 、鍵kとすると、

$$f(p', k) = \text{Rand}(p', k)$$

つまり、 $p'$ やkに対して出力 $f(p', k)$ がランダムに対応する。

③ 入力のビット変化量を $\Delta p'$ 、鍵のビット変化量を $\Delta k$ とすると、

$$f(\Delta p', \Delta k) = \text{Rand}(\Delta p', \Delta k)$$

つまり、 $\Delta p'$ や $\Delta k$ に対して、出力変化量もランダムに対応する。

④ 暗号化は、

$$c = f(p')$$

で表せ、復号化は、

$$p' = f(c) \text{ である。}$$

⑤ 入力 $p'$ や出力cが知られている時、鍵Kの虱み潰し検査により鍵Kは解読されるが、 $p'$ 又はcの一方が未知の時、その一方と鍵Kは解読不可能な構造をとる。

アルゴリズムfとしては、DES方式やFEAL方式を用いることができる。

また、アルゴリズムhは、以下のような構造を持つものである。

①  $h = h^{-1}$ 又は $h^{-1}$ が存在する。

② 入力pのビット変化量 $\Delta p$ に対して、

$$h(p, k) \neq \text{Rand}(p, K)$$

$$h(\Delta p, \Delta k) \neq \text{Rand}(\Delta p, \Delta K)$$

であっても良い。

③ 入力pと出力 $p'$ の両方が知られた時、gの構造が知られても良いが、gの構造やg内の変数が知られない時、pと $p'$ の一方が知られても、他方を知ることができない。

このような構造とした場合の暗号化は、

$$c = h^{-1}(f(h(p))) \text{ であり、}$$

復号化は、

$$p = h^{-1}(f(h(p))) \text{ である。つまり、インボリューション構造をなす。}$$

ここで、 $f(h(p))$ に注目してみると、

$$f(h(p)) = f(p, K, k)$$

つまり、独立変数p, K, kの関数であり、然も、

$$f(p, K, k) = \text{Rand}(p, K, k)$$

$$f(\Delta p, \Delta K, \Delta k)$$

$$= \text{Rand}(\Delta p, \Delta K, \Delta k)$$

が成立している。

つまり、この構造では、Kやkの虱み潰し検査以外の解読は不可能である。Kやkのビット長をBとすると、この方法の解読回数は、虱み潰しを行った場合、 $2^{B-1}$ 回となる。すなわち、 $B = 64$ とした場合には、 $2^{63}$ 回の演算が必要になり、 $1 \mu s$ の演算速度で、 $10^6$ の並列処理を行ったとしても、平均約 $10^{18}$ 年処理時間がかかることになり、十分安全な暗号と言える。

このようなアルゴリズムで符号化を行う場合の

**THIS PAGE BLANK (USPTO)**

アルゴリズム $h$ としては、例えばEX-ORを用いることができる。すなわち、第11図に示すように、このようなアルゴリズムの符号化は、EX-OR回路71及び72と、アルゴリズム $f$ の暗号化をし行う暗号化手段72とにより構成できる。

b2. 強い暗号化処理手段の前段と後段に弱い暗号化手段を配して、暗号化強度を強化させる構成の実施例

第12図は、このように強い暗号化処理手段の前段と後段に弱い暗号化手段を配して、暗号化強度を強化させて通信を行う場合の実施例を示すものである。

第12図において、通信を行うコンピュータシステム81には、鍵(K及び $k$ )82及びその鍵32を管理するための鍵管理手段83が設けられている。この鍵82は、鍵管理手段83により、例えばソフトウェアで管理されている。

暗号/復号器85及びモデム86は、コントロールライン87を介してコンピュータシステム8

1と結ばれている。

コンピュータシステム81から回線84を介してデータを送信する場合には、コンピュータシステム81からのデータが暗号/復号器85に供給される。そして、暗号/復号器85により、アルゴリズム $h$ により弱い暗号化がなされた後、アルゴリズム $f$ により強い暗号化がなされ、更に、アルゴリズム $h^{-1}$ により、弱い暗号化がなされ、この暗号化されたデータがモデム86を介して回線84に出力される。

回線84を介して伝えられてきたデータをコンピュータシステム81で受信する場合には、回線84を介して伝えられてきたデータがモデム86を介して暗号/復号器85に供給される。暗号/復号器85で、アルゴリズム $h$ による復号がなされ、更に、アルゴリズム $f$ による復号がなされ、更に、アルゴリズム $h^{-1}$ による復号がなされ、送られてきたデータが解読される。

この解読されたデータがコンピュータシステム81に供給される。

#### (発明の効果)

この発明によれば、DES方式やFEAL方式等の強い暗号化/復号化回路に、簡単なアルゴリズムの処理を行う回路を付加するだけで、暗号強度を強化でき、並列処理コンピュータにより演算速度が向上された場合でも、解読の危険性がなくなり、データの安全性を高めることができる。

#### 4. 図面の簡単な説明

第1図は構造化鍵を用いた暗号化の説明に用いるブロック図、第2図は構造化鍵を用いた暗号の復号化の説明に用いるブロック図、第3図は構造化鍵としてEX-ORを用いた場合の説明に用いるブロック図、第4図は構造化鍵としてEX-ORを用いた場合の解読の説明に用いるブロック図、第5図はビット転置器の説明に用いる略線図、第6図は構造化鍵としてビット転置器を用いた場合の暗号化処理の説明に用いるブロック図、第7図は構造化鍵としてビット転置器を用いた場合の復号化処理の説明に用いるブロック図、第8図は構

造化鍵を用いた場合の実施例のブロック図、第9図は強い暗号化処理の前後に弱い暗号化処理を配置するようにした暗号化の説明に用いるブロック図、第10図は強い暗号化処理の前後に弱い暗号化処理を配置するようにした暗号の復号化の説明に用いるブロック図、第11図は強い暗号化処理の前後に弱い暗号化処理を配置するようにした暗号化において弱い暗号化処理にEX-ORを用いるようにした例の説明に用いるブロック図、第12図は強い暗号化処理の前後に弱い暗号化処理を配置するようにした場合の実施例のブロック図である。

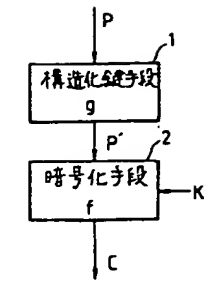
#### 図面における主要な符号の説明

31, 81: コンピュータシステム、

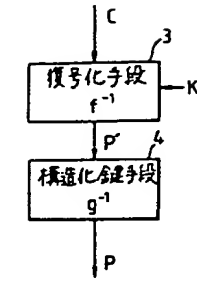
32: 構造化鍵手段、35, 85: 暗号/復号器。

代理人 弁理士 杉 浦 正 知

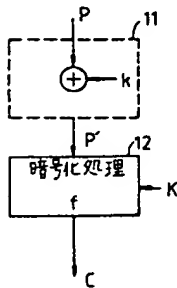
**THIS PAGE BLANK (USPTO)**



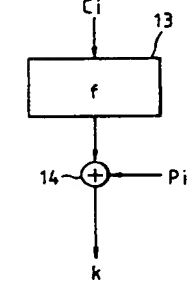
構造化鍵を用いた暗号化  
第1図



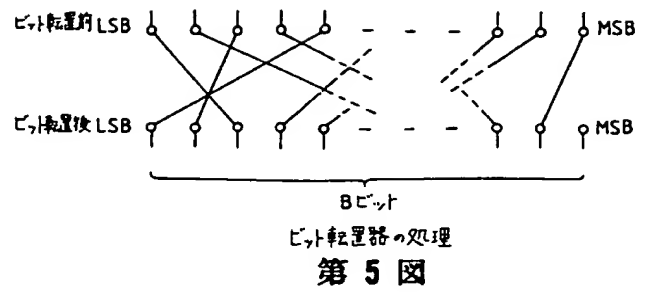
構造化鍵を用いた復号化  
第2図



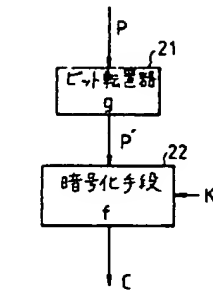
構造化鍵としてEX-ORを用いた  
場合の説明  
第3図



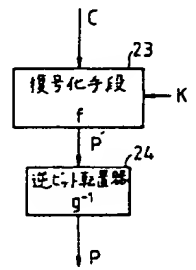
構造化鍵としてEX-ORを用いた  
場合の復号の説明  
第4図



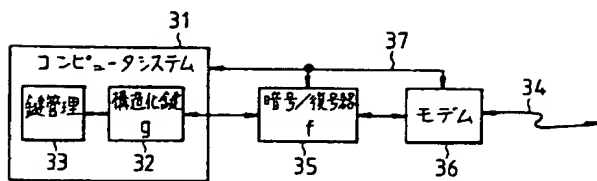
ビット転置器の処理  
第5図



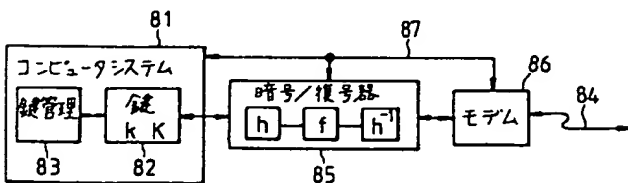
ビット転置器を用いた場合の  
暗号化処理  
第6図



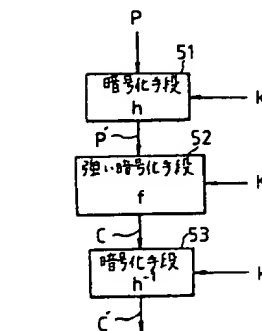
ビット転置器を用いた場合の  
復号化処理  
第7図



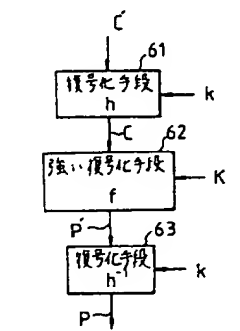
構造化鍵を用いた場合の実施例  
第8図



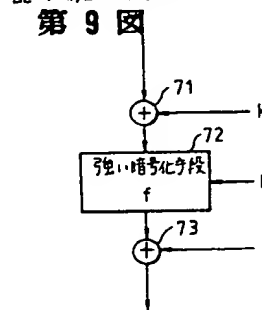
強い暗号化の前後に弱い暗号化を  
配した場合の実施例  
第12図



強い暗号化の前後に弱い暗号化を  
配する場合の暗号化の例  
第9図



強い暗号化の前後に弱い暗号  
化を配する場合の復号化の例  
第10図



強い暗号化手段の前後にEX-ORE  
配した例  
第11図

**THIS PAGE BLANK (USPTO)**